REMARKS/ARGUMENTS

Applicants appreciate the allowance of claims 7-25 and 32-50, and the indication that claims 4-6 and 29-31 would be allowable if re-written in independent form.

On page 3 of the Official Action, claims 1-3 and 26-28 were under 35 U.S.C. 103(a) as being unpatentable over "Network File Systems (NFS) version 4 Protocol" (hereinafter "NFS_V4") in view of Kirnos (U.S. Pat. App Pub. 2002/0188667). Applicants respectfully traverse.

Paragraph 6 on page 3 of the Official Action cites NFS_V4 for disclosing a file server system having a clock for producing a clock time and a processor for servicing client requests for access to a file, the processor having a timer for measuring a time interval. NFS_V4 mentions timers at the end of Sec. 3.1 (on page 21 of 245) with respect to Ports and Transports. NFS mentions "current_time" at the bottom of page 96 of 245 in processing in which the server uses its current time to construct attribute values for "time_metadata" and "time_modify".

Paragraph 6 on page 3 of the Official Action cites NFS_V4, page 96, lines 1-12 for the processor obtaining the clock time from the clock, and beginning measurement of the time interval with the timer. Page 96 of NFS_V4 says: "The server then uses its current time to construct attribute values for time_metadata and time_modify" and in particular "time_modify = time_metadata = current_time." (See the last line at the bottom of page 96 of 245.) However, page 96 of NFS_V4 does not mention measurement of any time interval with a timer. Nor is the

file modification time determined as a function of any time interval measured by any timer because the file modification time is determined as equal to the clock time.

Paragraph 6 on page 3 of the Official Action cites NFS_V4, page 96, lines 1-12 for "the processor responding to a request from a client for an asynchronous write to the file by performing an asynchronous write operation with respect to the file."   This page does not mention asynchronous writes, but NFS_V4 mentions asynchronous writes on page 89 of 245 in section 9.3.2 regarding Data Caching and File Locking.

Paragraph 6 on pages 3-4 of the Official Action says: "In the same field of endeavor, Kirnos discloses determining a file-modification time that is a function of the clock time having been obtained from the clock and the time interval measured by the timer, the file modification time indicating a time of modification of the file by the asynchronous write operation (i.e., file modification time = start time + f, where f is the time interval between t=0 and t=f for performing one or more operations that alters the file)(Kirnos, paragraphs [0059] and [0094])." However, Kirnos paragraph [0059] discloses that a working version 50 and a comparison file 70 are made at a time t=0, and a user makes a request to synchronize the working version 50 with the file system 40 at a time t=i. Kirnos paragraph [0094] discloses that a working version 50 and a comparison file 70 are made at the time t=0, and a user makes a request to synchronize the working version 50 with the file system 40 at a time t=f. Neither of these paragraphs discloses use of a timer nor discloses how the system of Kirnos might determine a file modification time. The drafter of the Kirnos patent has simply chosen to designate the time of occurrence of a first

event as t=0 and to designate the time of occurrence of a second later event as t=a for the sake of illustrating a sequential ordering of the events.

Instead of using a timer, Kirnos paragraph [0049] discloses that when a working version 50 is generated by downloading to the first computer 10, the modification time is recorded by an operating system (or other application) on the first terminal. "The modification time is a value associated with the working items to mark the last instant that the item was edited or created. The modification time can therefore change after working version 50 is downloaded from the file system 40." Kirnos paragraph [0112] also says: "As mentioned, both the creation time and the modification time are time values that are automatically recorded by operating systems such as WINDOWS." Kirnos paragraph [0123] further says: "An embodiment provides that modification time of a copy is the same as the modification time of its original."

Paragraph 6 on page 4 of the Official Action concludes that it would have been obvious to incorporate the features of determining a file-modification time of the file, as disclosed by Kirnos, into the teachings of NFS_V4. However, this would not result in the subject matter of applicants' claim 1 because each of NFS_V4 and Kirnos teaches that the file modification time is determined as a function of a clock time by setting the file modification time to the current clock time or recording the current clock time as the file modification time when the file is modified. Neither NFS_V4 nor Kirnos suggests determining the modification time of a file as a function of a time interval in addition to a clock time, nor more particularly a clock time and a time interval having the relationship specified in the first step of applicants' claim 1 with respect to the

processor responding to a request from a client for an asynchronous write in the second step of applicants' claim 1.

With respect to the applicants' dependent claim 2, paragraph 7 on page 4 of the Official Action cites Kirnos, paragraphs [0059] and [0094] for suggesting that the file modification time = start time + f wherein f is the time interval between t=0 and t=f for performing one or more operations that alters the file. However, the fact that a working version 50 and a comparison file 70 are made at the time designated as t=0, and a user makes a request to synchronize the working version 50 with the file system 40 at a time designated as t=i or t=f, as disclosed in Kirnos, paragraphs [0059] and [0094], does not suggest that a clock time is obtained from a clock at the time t=0 and a timer is started at the time t=0, and at a time t=f, a file modification time is determined as a function of the clock time obtained from the clock at t=0 and the time interval measured by the timer. Instead, Kirnos teaches that the modification time for a file is obtained from the current time when the file is modified.

With respect to the applicants' dependent claim 3, paragraph 8 on page 4 of the Official Action cites NFS_V4, page 33, sections 5.1-5.2 and Kirnos, paragraph [0112] for suggesting that the processor acknowledges the request from the client for an asynchronous write to the file by returning to the client the file-modification time. However, NFS_V4, page 33, sections 5.1-5.2 teaches that a client may ask for the file modification time attribute by setting a bit in the GETATTR request and the server must return its value. A GETATTR (i.e., get attribute) request is not an asynchronous write request. Kirnos, paragraph [0112] teaches that the modification time is automatically recorded by operating systems such as WINDOWS. Therefore, the cited

portions of NFS_V4 and Kirnos do not suggest the specific limitations expressly recited in applicants' dependent claim 3.

Paragraph 9 on page 5 of the Official Action rejected the applicants' file server claims 26-28 under the same rationale as the rejection of applicants' method claims 1-3. Applicants respectfully traverse, for the reasons given above in response to the rejection of the applicants' claims 1-3.

In view of the above, it is respectfully submitted that the application is in condition for allowance. Reconsideration and early allowance are earnestly solicited.

Respectfully submitted,

/ *Richard C. Auchterlonie* /

Richard C. Auchterlonie, Reg. No. 30,607

NOVAK DRUCE & QUIGG, LLP
1000 Louisiana, 53rd Floor
Houston, TX 77002
713-571-3460